

On the number of segments needed in a piecewise linear approximation

Christopher L. Frenzen^a, Tsutomu Sasao^b and Jon T. Butler^{c*}.

^aDepartment of Applied Mathematics, Naval Postgraduate School, Monterey, CA, 93943-5216, U.S.A., cfrenzen@nps.edu .

^bDepartment of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka, 820-8502, JAPAN, sasao@cse.kyutech.ac.jp .

^cDepartment of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, 93943-5121, U.S.A., jon_butler@msn.com .

Abstract

The introduction of high-speed circuits to realize an arithmetic function f as a piecewise linear approximation has created a need to understand how the number of segments depends on the interval $a \leq x < b$ and the desired approximation error ε . For the case of optimum non-uniform segments, we show that the number of segments is given as $s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}$, ($\varepsilon \rightarrow 0$), where $c = \frac{1}{4} \int_a^b \sqrt{|f''(x)|} dx$. We also show that, if the segments have the same width (to reduce circuit complexity), then the number of segments is given by $s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}$, ($\varepsilon \rightarrow 0$), where $c = \frac{(b-a)\sqrt{|f''|_{\max}}}{4}$.

Keywords: piecewise linear approximation; numeric function generators

1. Introduction

A **numeric function generator (NFG)** is a logic circuit that realizes an arithmetic function like $f(x) = \sin x$ over some specified interval $a \leq x < b$. We consider the numeric function generator shown in Fig. 1. This architecture realizes a given function as a set of segments or pieces, where f is approximated in each segment by the linear equation $f(x) \approx c_1x + c_0$. The values of c_1 and c_0 are stored in the Coefficients Memory shown in Fig. 1 in a location whose address is specified by the Segment Index Encoder. In designing this circuit, one partitions the interval $a \leq x < b$ into segments, where the coefficients c_1 and c_0 are the same in each segment and approximate the function to within some specified error. It is known [9] that the Segment Index Encoder is tractably realized.

This circuit is said to realize a **non-uniform segmentation** because, in general, the segments have different widths. Indeed, we will choose the segment widths as large as possible so that the approximation is still less than or equal to within the given approximation error. In this way, we produce a segmentation that has a few segments as possible.

*Corresponding author at: Department of Electrical and Computer Engineering, Naval Postgraduate School, Code EC/Bu, Monterey, CA, 93943-5121.

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE MAY 2010		2. REPORT TYPE		3. DATES COVERED	
4. TITLE AND SUBTITLE On the number of segments needed in a piecewise linear approximation			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School, Department of Electrical and Computer Engineering, Monterey, CA, 93943			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The introduction of high-speed circuits to realize an arithmetic function f as a piecewise linear approximation has created a need to understand how the number of segments depends on the interval $a \leq x < b$ and the desired approximation error ϵ. For the case of optimum non-uniform segments, we show that the number of segments is given as $s(\epsilon) \leq pc$, ($\epsilon \neq 0$), where $c = 1 + 4R \int_a^b p f(x) dx$. We also show that, if the segments have the same width (to reduce circuit complexity), then the number of segments is given by $s(\epsilon) \leq pc$, ($\epsilon \neq 0$), where $c = (b-a) \int_a^b p f(x) dx$.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 13	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

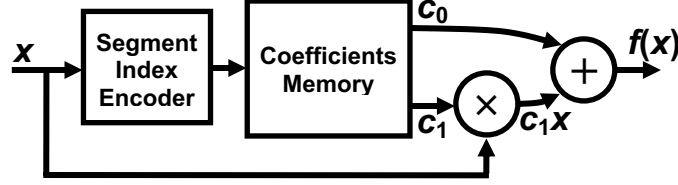


Figure 1. Architecture of a Numerical Function Generator Using Piecewise Linear Approximation and Non-Uniform Segmentation.

In contrast, we also consider a **uniform segmentation**. In this case, all segments have equal width. If the segment widths are 2^m , then the segment index encoder can be removed and the higher order $n - m$ bits used to drive the address of the Segment Index Encoder, where n is the number of bits to encode x . In this case, the circuit is likely to be smaller and faster.

Up to this point, we have not had an analytical tool to predict the size of the Coefficients Memory as a function of the function realized, the domain, and the required approximation error. That is, our understanding of its size has only been through experimental results; i.e. specific implementations [5,7–10]. We derive an expression for the number of segments (size of the Coefficients Memory) for both the non-uniform and the uniform case.

2. Non-Uniform Approximation With Unrestricted Slope

Let f be a three times continuously differentiable function defined on the domain $[a, b]$. In the case of unrestricted slope, our algorithm proceeds by generating a segmentation $\{x_0, x_1, \dots, x_n\}$ of $[a, b]$ with the property that, in each of the n segments $[x_i, x_{i+1}]$, $i = 0, \dots, n - 1$, the chord between x_i and x_{i+1} produces a linear approximation $(c_1x + c_0)$ to f within a previously specified approximation error ε :

$$|f(x) - c_1x - c_0| \leq \varepsilon, \quad x \in [x_i, x_{i+1}].$$

In the example of the $\sin(\pi x)$ function, experimental results show that, for a specified approximation error ε , the segmentation algorithm, in the case of unrestricted slope, determines a number of segments s that is proportional to $1/\sqrt{\varepsilon}$. We now show that this is a general result for a large set of functions.

Specifically, we give an asymptotic approximation for the number of segments $s(\varepsilon)$ needed to approximate a given function $f(x)$ to within a given approximation error ε . We say that $t(\varepsilon)$ is an asymptotic approximation to $s(\varepsilon)$, expressed as $s(\varepsilon) \sim t(\varepsilon)$, if $\lim_{\varepsilon \rightarrow 0^+} s(\varepsilon)/t(\varepsilon) = 1$. In our use of this, $s(\varepsilon)$ is the exact number of segments, while $t(\varepsilon)$ is an approximation to $s(\varepsilon)$, which has a simple form. Intuitively, we expect the number of segments to increase as the approximation error ε decreases. We seek to determine this relationship, since it provides insight into how hardware complexity depends on the approximation error.

In what follows, we divide the domain $[a, b]$ into two sets depending of the value of

$f''(x)$. Let

$$A_\varepsilon = \{x \in [a, b] : |f''(x)| \leq \sqrt{\varepsilon}\} \quad \text{and} \quad (1)$$

$$B_\varepsilon = \{x \in [a, b] : |f''(x)| > \sqrt{\varepsilon}\}, \quad (2)$$

where $\varepsilon > 0$. Note that 1) A_ε is a closed set; 2) B_ε is an open set, 3) $A_\varepsilon \cap B_\varepsilon = \emptyset$; and 4) $A_\varepsilon \cup B_\varepsilon = [a, b]$.

Theorem 1. *Consider a piecewise linear approximation of f on the domain $[a, b]$ that is accurate to within ε , using a piecewise linear segmentation. Let f be three times continuously differentiable on $[a, b]$. Then, $s(\varepsilon)$, the number of segments in an optimum segmentation of $[a, b]$, satisfies the following asymptotic approximation:*

$$s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}, \quad (\varepsilon \rightarrow 0), \quad (3)$$

where

$$c = \frac{1}{4} \int_a^b \sqrt{|f''(x)|} dx. \quad (4)$$

Proof See Appendix.

For example, if we take $f(x) = \sin \pi x$ on $[0, \frac{1}{2}]$, we find

$$s(\varepsilon) \sim \frac{\pi \int_0^{1/2} \sqrt{\sin \pi x} dx}{4\sqrt{\varepsilon}}. \quad (5)$$

Using MAPLE to numerically evaluate the integral in (5) yields

$$s(\varepsilon) \sim \frac{0.2995}{\sqrt{\varepsilon}}.$$

The following corollary of Theorem 1 relaxes the conditions on the end points of the domain.

Corollary 1. *Let f be three times continuously differentiable on the open interval (a, b) and $\sqrt{f''}$ be improperly Riemann integrable² on the closed interval $[a, b]$, with integrable singularities at the endpoints a or b . Then, $s(\varepsilon)$, the number of segments in an optimum segmentation of $[a, b]$, satisfies the following asymptotic approximation:*

$$s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}, \quad (\varepsilon \rightarrow 0), \quad (6)$$

where

$$c = \frac{1}{4} \int_a^b \sqrt{|f''(x)|} dx. \quad (7)$$

²The function f is improperly Riemann integrable on $[a, b]$ if f is Riemann integrable on every open subinterval (c, d) of $[a, b]$, and the function f becomes unbounded in neighborhoods of a or b , and $\lim_{c \rightarrow a, d \rightarrow b} \int_c^d f(x) dx$ exists.

The significance of Corollary 1 is that we can obtain an asymptotic approximation to the number of segments even when the function has singularities at the endpoints of the interval over which the approximation occurs.

Table 1 shows the asymptotic approximations to the number of segments for 14 functions. These values were derived from Theorem 1 and Corollary 1, and are shown in the third column labeled **Non-uniform**. The two functions, \sqrt{x} and the entropy function, $-x \log_2 x - (1-x) \log_2(1-x)$, on which we applied Corollary 1, are among the functions with the largest number of segments. As we will see later, these functions are especially expensive to approximate using uniform segmentation. The fourth column labeled **Uniform** will be discussed in Section 3. The fifth (rightmost) column, labeled **Non-Uni/Uni**, shows the ratio of the number of non-uniform segments needed compared to the number of uniform segments needed as a percentage, when the specified approximation error ε is small. For example, for 2^x , non-uniform segmentation uses 84% of the segments needed by uniform segmentation. For \sqrt{x} and $-(x \log_2 x + (1-x) \log_2(1-x))$, the fifth column contains (0%), which shows that the ratio of segments in non-uniform segmentation is vanishingly small compared to the number of segments needed in the uniform approximation as $\varepsilon \rightarrow 0$. We discuss the derivations for the number of segments needed for uniform segmentation in Section 3.

Table 1
Number of Segments for Non-Uniform and Uniform Segmentations

Function $f(x)$	Domain of x	Non-Uniform $s \sim$	Uniform $s \sim$	Non-Uni /Uni =
2^x	$[0, 1)$	$\frac{0.2071}{\sqrt{\varepsilon}}$	$\frac{0.2451}{\sqrt{\varepsilon}} = \frac{\ln 2}{\sqrt{8}\sqrt{\varepsilon}}$	84%
$1/x$	$[1, 2)$	$\frac{0.2071}{\sqrt{\varepsilon}}$	$\frac{0.3536}{\sqrt{\varepsilon}} = \frac{1}{\sqrt{8}\sqrt{\varepsilon}}$	59%
\sqrt{x}	$[0, 2)$	$\frac{0.5946}{\sqrt{\varepsilon}}$	$\frac{0.0313}{\varepsilon^2} = \frac{1}{32\varepsilon^2}$	(0%)
$1/\sqrt{x}$	$[1, 2)$	$\frac{0.1378}{\sqrt{\varepsilon}}$	$\frac{0.2165}{\sqrt{\varepsilon}} = \frac{\sqrt{3}}{8\sqrt{\varepsilon}}$	64%
$\log_2(x)$	$[1, 2)$	$\frac{0.2081}{\sqrt{\varepsilon}}$	$\frac{0.3003}{\sqrt{\varepsilon}} = \frac{1}{4\sqrt{\ln 2}\sqrt{\varepsilon}}$	69%
$\ln x$	$[1, 2)$	$\frac{0.1733}{\sqrt{\varepsilon}}$	$\frac{0.2500}{\sqrt{\varepsilon}} = \frac{1}{4\sqrt{\varepsilon}}$	69%
$\sin(\pi x)$	$[0, \frac{1}{2})$	$\frac{0.2995}{\sqrt{\varepsilon}}$	$\frac{0.3927}{\sqrt{\varepsilon}} = \frac{\pi}{8\sqrt{\varepsilon}}$	76%
$\cos(\pi x)$	$(0, \frac{1}{2})$	$\frac{0.2995}{\sqrt{\varepsilon}}$	$\frac{0.3927}{\sqrt{\varepsilon}} = \frac{\pi}{8\sqrt{\varepsilon}}$	76%
$\tan(\pi x)$	$[0, \frac{1}{4})$	$\frac{0.2005}{\sqrt{\varepsilon}}$	$\frac{0.3927}{\sqrt{\varepsilon}} = \frac{\pi}{8\sqrt{\varepsilon}}$	51%
$\sqrt{-\ln(x)}$	$[\frac{1}{256}, \frac{1}{4})$	$\frac{0.5961}{\sqrt{\varepsilon}}$	$\frac{6.9226}{\sqrt{\varepsilon}} = \frac{15^{\frac{3}{4}} \sqrt{(\ln 256)^{-\frac{1}{2}} - \frac{1}{2}} (\ln 256)^{-\frac{3}{2}}}{\sqrt{2}\sqrt{\varepsilon}}$	9%
$\tan^2(\pi x) + 1$	$[0, \frac{1}{4})$	$\frac{0.4200}{\sqrt{\varepsilon}}$	$\frac{0.7854}{\sqrt{\varepsilon}} = \frac{\pi}{4\sqrt{\varepsilon}}$	53%
$-(x \log_2 x + (1-x) \log_2(1-x))$	$(0, 1)$	$\frac{0.9434}{\sqrt{\varepsilon}}$	$\frac{0.2654}{\varepsilon} = \frac{1}{2e \ln 2 \varepsilon}$	(0%)
$\frac{1}{1+e^{-x}}$	$[0, 1)$	$\frac{0.0550}{\sqrt{\varepsilon}}$	$\frac{0.0754}{\sqrt{\varepsilon}} = \frac{\sqrt{e^1+e^{-1}}}{e^1+2+e^{-1}} \frac{1}{4\sqrt{\varepsilon}}$	73%
$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	$[0, \sqrt{2}]$	$\frac{0.1452}{\sqrt{\varepsilon}}$	$\frac{0.2233}{\sqrt{\varepsilon}} = \frac{\sqrt{2}}{4(2\pi)^{1/4}\sqrt{\varepsilon}}$	65%

3. Uniform Approximation With Unrestricted Slope

In this part, we consider two ways to determine the number of segments needed in the case when a completely free choice of slope is used with uniform segmentation. The first approach is a direct computation, which can be applied to all functions considered in this paper, and the second is an asymptotic approximation that applies to a majority of the functions.

The first approach is illustrated as follows. Consider a uniform segmentation of a function $f(x)$ from $x = a$ to $x = b$. Consider a segment beginning at $x = \alpha$ and ending at $x = \beta$, where $\alpha < \beta$ near a point in the domain $[a, b]$, where $|f''(x)|$ is maximum. It is at this point that the maximum error between the function and its linear piecewise approximation occurs. That is, if we choose the segment width to be small enough at this point so that the error is equal to the specified approximation error ε , then that small width for all segments will be sufficient to achieve a maximum error ε in all segments. Therefore, the number of segments s , to achieve an approximation error with uniform segmentation is

$$s = \left\lceil \frac{b - a}{\beta - \alpha} \right\rceil. \quad (8)$$

Consider a piecewise linear approximation, $f_{\text{pl}}(x)$ to $f(x)$ of the form $f_{\text{pl}}(x) = (f(\beta) - f(\alpha))\frac{x-\alpha}{\beta-\alpha} + f(\alpha)$. The error due to the approximation can be viewed as $|f(x) - f_{\text{pl}}|$. Note that $|f(x) - f_{\text{pl}}|$ is 0 at $x = \alpha$ and at $x = \beta$. However, we will approximate $f(x)$ in the domain $[\alpha, \beta]$ by adding a constant to f_{pl} so that the maximum error in the domain $[\alpha, \beta]$ is no greater than $\frac{1}{2}|f(x) - f_{\text{pl}}|$. As a result, the error function $e(x)$ for the domain $[\alpha, \beta]$ is $\frac{1}{2}(f(x) - f_{\text{pl}})$. Substituting for f_{pl} , yields

$$e(x) = \frac{1}{2} \left[[f(x) - f(\alpha)] - [f(\beta) - f(\alpha)] \frac{x - \alpha}{\beta - \alpha} \right]. \quad (9)$$

To illustrate, consider the function $f(x) = \sqrt{x}$ in the domain $[0, 2]$. $f''(x)$ becomes unbounded near $x = 0$. Thus, we choose $\alpha = 0$, and from (9), we have

$$e(x) = \frac{1}{2} \left[\sqrt{x} - \sqrt{\beta} \frac{x}{\beta} \right].$$

By differentiating $e(x)$ with respect to x , we find that the maximum error occurs at $x = \frac{\beta}{4}$. At this value, the maximum e , e_{max} is $\frac{\sqrt{\beta}}{8}$. We choose this value to be ε . That is, $e_{\text{max}} = \varepsilon$, and, so $\frac{\sqrt{\beta}}{8} = \varepsilon$. Substituting this into (8) yields

$$s = \left\lceil \frac{1}{32\varepsilon^2} \right\rceil. \quad (10)$$

Thus, as ε decreases (improves), the number of segments needed for a uniform segmentation increases as the inverse of the square of ε . This results from the fact that f'' becomes unbounded near the endpoint 0. A similar analysis can be applied to the entropy function, and in fact can be generalized.

The number of segments needed for uniform segmentation can be computed for all of the target functions. Column 4 of Table 1, labeled **Uniform** shows the expressions of the number of segments required in a uniform approximation for the various functions. For all but two of the functions, the expressions have the form $\frac{c}{\sqrt{\varepsilon}}$, which are the same form as the asymptotic approximation for the number of segments required in a non-uniform segmentation. For two functions, \sqrt{x} and $-(x \log_2 x + (1-x) \log_2(1-x))$, the asymptotic approximation for the number of segments is $\frac{c}{\varepsilon^2}$ and $\frac{c}{\varepsilon}$, respectively. The right column of Table 1, labeled **Non-Uni/Uni** shows the ratio of segments required in a non-uniform segmentation to the number of segments required in a uniform segmentation (expressed as a percentage). For some functions, like 2^x , $\sin(\pi x)$, and $\cos(\pi x)$, this is high, 84%, 76%, and 76%, respectively. For such functions, there is a small penalty for using a uniform segmentation. For other functions, like $\sqrt{-\ln(x)}$, this percentage is low, 9%, and the penalty is high. A similar statement is true of \sqrt{x} and $-(x \log_2 x + (1-x) \log_2(1-x))$, where the percentage number of segments is expressed as (0%).

It is interesting that, when the domain for the \sqrt{x} function is reduced to $[\frac{1}{2}, 2]$, the number of segments is given by $s \sim \frac{0.3153}{\sqrt{\varepsilon}}$.

The second approach to determining the number of segments requires the second derivative $f''(x)$ to be bounded over the domain of approximation. We have

Theorem 2. *Consider a piecewise linear approximation of a function $f(x)$ on the domain $[a, b]$ with a specified approximation error ε or less using uniform segmentation. Let the absolute value of the second derivative $|f''(x)|$ of $f(x)$ on the domain $[a, b]$ be finite. Then, the number of segments s is*

$$s \sim \frac{c}{\sqrt{\varepsilon}}, \quad (11)$$

where

$$c = \frac{(b-a)\sqrt{|f''|_{\max}}}{4}, \quad (12)$$

where $|f''|_{\max}$ is the maximum of the absolute value of $f''(x)$ over the domain $[a, b]$.

Proof See Appendix.

The right column of Table 1 shows the results of Theorem 2. Specifically, all functions in this table except \sqrt{x} and $-(\log_2 x + (1-x) \log_2(1-x))$ satisfy the restriction that $|f''(x)|$ is finite. For example, for $\sin(\pi x)$, $|f''(x)|_{\max} = \pi^2$. Therefore, for this function, (12) yields $s \sim \frac{\pi}{8\sqrt{\varepsilon}}$, which agrees with Table 1.

4. Concluding Remarks

As a result of our analysis, we have an understanding of how the hardware complexity, as measured by the number of segments, depends on the specified precision ε . Our results

also show that, for some functions, it is reasonable to use uniform segmentation, thus eliminating the segment index encoder. For such functions, our results validate the past research on uniform segmentation. For example, with the $\sin(\pi x)$ and $\cos(\pi x)$ functions approximated using unrestricted slope, $\frac{0.3927}{\sqrt{\varepsilon}}$ segments are needed for uniform segmentation, while no more than $\frac{0.2995}{\sqrt{\varepsilon}}$ segments are needed for non-uniform segmentation. This is about 31% more segments. The penalty is substantial, but if memory is inexpensive, and speed is essential, this may be a welcome tradeoff.

Table 2 shows the number of segments needed in the case of non-uniform and uniform segmentation for functions approximated using unrestricted slope as calculated in Sections 4 and 5, respectively. Table 2 shows the number of segments needed for four precisions, 8, 16, 32, and 64 bits ³.

Table 2

Number of Segments for Non-Uniform and Uniform Segmentation For Four Precisions, 8, 16, 24, and 32 Bits.

Function $f(x)$	Inter- val x	Non-Uniform				Uniform			
		8	16	32	64	8	16	32	64
2^x	$[0, 1)$	4	75	19,195	1.26×10^9	6	89	22,717	1.49×10^9
$1/x$	$[1, 2)$	4	75	19,195	1.26×10^9	8	128	32,773	2.15×10^9
\sqrt{x}	$[0, 2)$	10	216	55,109	3.61×10^9	8,206	5.38×10^8	2.31×10^{18}	4.26×10^{37}
$1/\sqrt{x}$	$[1, 2)$	3	50	12,772	8.37×10^8	5	79	20,066	1.32×10^9
$\log_2(x)$	$[1, 2)$	4	75	19,228	1.26×10^9	7	109	27,833	1.82×10^9
$\ln x$	$[1, 2)$	3	63	16,062	1.05×10^9	6	91	23,171	1.52×10^9
$\sin(\pi x)$	$[0, \frac{1}{2})$	5	109	27,759	1.82×10^9	9	143	36,397	2.39×10^9
$\cos(\pi x)$	$(0, \frac{1}{2})$	5	109	27,759	1.82×10^9	9	143	36,397	2.39×10^9
$\tan(\pi x)$	$[0, \frac{1}{4})$	4	73	18,583	1.22×10^9	9	143	36,397	2.39×10^9
$\sqrt{-\ln(x)}$	$[\frac{1}{256}, \frac{1}{4})$	10	216	55,248	3.62×10^9	157	2,507	641,600	4.20×10^{10}
$\tan^2(\pi x) + 1$	$[0, \frac{1}{4})$	7	153	38,927	2.55×10^9	18	285	72,793	4.77×10^9
$-(x \log_2 x + (1-x) \log_2(1-x))$	$(0, 1)$	16	342	87,437	5.73×10^9	136	34,787	2.28×10^9	9.79×10^{18}
$\frac{1}{1+e^{-x}}$	$[0, 1)$	1	20	5,096	3.34×10^8	2	28	6,989	4.58×10^8
$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	$[0, \sqrt{2}]$	3	53	13,458	8.82×10^8	6	81	20,696	1.36×10^9

From Table 2, we can make conclusions about the feasibility of realizing the various functions. Specifically, we can see that for 64 bit precision, very large memory size is needed in all cases. For 32 bit precision, both uniform and non-uniform segmentation yield feasible realizations, except for \sqrt{x} and $-(x \log_2 x + (1-x) \log_2(1-x))$. Also, the memory required to realize $\sqrt{-\ln(x)}$ is quite large compared to that required for non-uniform segmentation. For 16 bit precision, all realizations of the functions are feasible, except for \sqrt{x} using uniform segmentation. For 8 bit precision, all realizations are feasible using either non-uniform and uniform segmentation. In general, for many functions,

³Assuming that the most significant bit is the coefficient of 2^{-1} , we choose the error, ε , to be one-half of the value of the least significant bit. For example, for 8 bit precision, we choose the error to be 2^{-9} . We substitute this for ε in the equations for the number of segments for non-uniform segmentation and uniform segmentation.

uniform segmentation is good, especially when the cost of memory is low.

Acknowledgment

This research is supported in part by the National Security Agency, Grants in Aid for Scientific Research of JSPS, and MEXT, and a grant of the Knowledge Cluster Project.

5. APPENDIX

In the following theorem, we make the following reasonable assumptions.

Assumption 1: As $\varepsilon \rightarrow 0$, the *measure* of $A_\varepsilon \rightarrow 0$.

Assumption 2: The open set B_ε is a finite union of open intervals.

Given these assumptions, we now state

Theorem 1 *Consider a piecewise linear approximation of f on the domain $[a, b]$ that is accurate to within ε , using a piecewise linear segmentation. Let f be three times continuously differentiable on $[a, b]$. Then, $s(\varepsilon)$, the number of segments in an optimum segmentation of $[a, b]$, satisfies the following asymptotic approximation:*

$$s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}, (\varepsilon \rightarrow 0),$$

where

$$c = \frac{1}{4} \int_a^b \sqrt{|f''(x)|} dx.$$

Proof Given $\varepsilon > 0$, divide the domain $[a, b]$ into segments with end points $\{x_0, x_1, \dots, x_s\}$, where $x_0 = a$ and $x_s = b$. Assume $\{x_0, x_1, \dots, x_s\}$ has the fewest segments such that all segments have an approximation error no greater than ε . Thus, for any segment, if we set

$$L_i(x) = f(x) - \left[\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} (x - x_i) + f(x_i) \right], \quad (13)$$

then

$$|L_i(x)| \leq 2\varepsilon,$$

for $x_i \leq x \leq x_{i+1}$, where $i = 0, 1, \dots, s-1$. Without loss of generality, since the segmentation is optimal, we can assume that, for all but perhaps one segment, there exists an x_i^* in (x_i, x_{i+1}) , for which $|L_i(x_i^*)| = 2\varepsilon$, for $i = 0, 1, \dots, s-1$. Now let

$$x_{\text{mid}}^{(i)} = \frac{x_i + x_{i+1}}{2}$$

and

$$\Delta_i = x_{i+1} - x_i$$

be respectively, the midpoint and length of the segment $[x_i, x_{i+1}]$ so that

$$x_i = x_{\text{mid}}^{(i)} - \frac{\Delta_i}{2}, \quad (14)$$

$$x_{i+1} = x_{\text{mid}}^{(i)} + \frac{\Delta_i}{2}, \quad (15)$$

and, for $x \in [x_i, x_{i+1}]$,

$$x = x_{\text{mid}}^{(i)} - \alpha_i(x) \frac{\Delta_i}{2},$$

where $-1 \leq \alpha_i(x) \leq 1$.

Apply Taylor's approximation to the terms in L_i in (13). This yields, after some algebra,

$$L_i(x) = \frac{1}{2} f''(x_{\text{mid}}^{(i)}) \frac{\Delta_i^2}{4} (\alpha_i^2 - 1) + O(\Delta_i^3).$$

Hence,

$$|L_i(x)| = \frac{\Delta_i^2}{8} |f''(x_{\text{mid}}^{(i)})| (|\alpha_i^2 - 1| + O(\Delta_i)),$$

for $i = 0, 1, \dots, s-1$ and $x_i \leq x \leq x_{i+1}$. Since $-1 \leq \alpha_i(x) \leq 1$,

$$\max_{\alpha_i(x) \in [-1, 1]} |\alpha_i(x)^2 - 1| = 1.$$

Therefore,

$$2\varepsilon = \max_{[x_i, x_{i+1}]} |L_i(x)| = \frac{\Delta_i^2}{8} |f''(x_{\text{mid}}^{(i)})| (1 + O(\Delta_i)). \quad (16)$$

Take the square root of both sides and sum over i from 0 to $s-1$:

$$\sum_{i=0}^{s-1} \sqrt{2\varepsilon} = \sum_{i=0}^{s-1} \frac{\Delta_i}{\sqrt{8}} \sqrt{|f''(x_{\text{mid}}^{(i)})|} (1 + O(\Delta_i)).$$

We now recognize that $x_{\text{mid}}^{(i)}$ must lie in either the set A_ε or the set B_ε , where

$$A_\varepsilon = \{x \in [a, b] : |f''(x)| \leq \sqrt{\varepsilon}\} \quad \text{and} \quad (17)$$

$$B_\varepsilon = \{x \in [a, b] : |f''(x)| > \sqrt{\varepsilon}\}, \quad (18)$$

for $\varepsilon > 0$.

Thus, we split the sum accordingly:

$$\begin{aligned} \sum_{i=0}^{s-1} \sqrt{2\varepsilon} &= \sum_{x_{\text{mid}}^{(i)} \in A_\varepsilon} \frac{\Delta_i}{\sqrt{8}} \sqrt{|f''(x_{\text{mid}}^{(i)})|} (1 + O(\Delta_i)) \\ &+ \sum_{x_{\text{mid}}^{(i)} \in B_\varepsilon} \frac{\Delta_i}{\sqrt{8}} \sqrt{|f''(x_{\text{mid}}^{(i)})|} (1 + O(\Delta_i)). \end{aligned} \quad (19)$$

The first sum on the right in (19) is small, $O(\varepsilon^{1/4})$. Also, since

$$|f''(x_{\text{mid}}^{(i)})| > \sqrt{\varepsilon}$$

for $x_{\text{mid}}^{(i)} \in B_\varepsilon$, (16) implies

$$2\varepsilon \geq \frac{\Delta_i^2}{8} |f''(x_{\text{mid}}^{(i)})| (1 + O(\Delta_i)) > \frac{\Delta_i^2}{8} \sqrt{\varepsilon} (1 + O(\Delta_i)). \quad (20)$$

It follows from (20) that $\Delta_i^2 = O(\sqrt{\varepsilon})$, and so $\Delta_i \rightarrow 0$ as $\varepsilon \rightarrow 0$. Now, (19) can be written as

$$s(\varepsilon) \sqrt{2\varepsilon} = O(\varepsilon^{1/4}) + \sum_{x_{\text{mid}}^{(i)} \in B_\varepsilon} \frac{\Delta_i}{\sqrt{8}} \sqrt{|f''(x_{\text{mid}}^{(i)})|} (1 + O(\Delta_i)). \quad (21)$$

Since $\Delta_i \rightarrow 0$ as $\varepsilon \rightarrow 0$, $A_\varepsilon \cup B_\varepsilon = [a, b]$, and the measure of A_ε tends to 0 as $\varepsilon \rightarrow 0$, it follows that the number of terms in the sum in (21) goes to infinity as $\varepsilon \rightarrow 0$, and so, by our previous assumptions, we can approximate the sum in (21) as a Riemann integral:

$$\begin{aligned} \sum_{x_{\text{mid}}^{(i)} \in B_\varepsilon} \frac{\Delta_i}{\sqrt{8}} \sqrt{|f''(x_{\text{mid}}^{(i)})|} (1 + O(\Delta_i)) &= \\ \frac{1}{\sqrt{8}} \int_{B_\varepsilon} \sqrt{|f''(x)|} dx &(1 + o(1)). \end{aligned}$$

Thus,

$$s(\varepsilon) \sqrt{2\varepsilon} = O(\varepsilon^{1/4}) + \frac{1}{\sqrt{8}} \int_{B_\varepsilon} \sqrt{|f''(x)|} dx (1 + o(1))$$

and

$$s(\varepsilon) = O(\varepsilon^{-1/4}) + \frac{1}{4\sqrt{\varepsilon}} \int_{B_\varepsilon} \sqrt{|f''(x)|} dx (1 + o(1)).$$

Since the measure of the set $A_\varepsilon \rightarrow 0$ as $\varepsilon \rightarrow 0$, a combination of the above results gives

$$s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}, \quad (\varepsilon \rightarrow 0),$$

where

$$c = \frac{1}{4} \int_a^b \sqrt{|f''(x)|} dx.$$

■

Theorem 2 Consider a piecewise linear approximation of a function $f(x)$ on the domain $[a, b]$ with a specified approximation error ε or less using uniform segmentation. Let the absolute value of the second derivative $|f''(x)|$ of $f(x)$ on the domain $[a, b]$ be bounded. Then, the number of segments s is

$$s \sim \frac{c}{\sqrt{\varepsilon}}, \quad (22)$$

where

$$c = \frac{(b-a)\sqrt{|f''|_{\max}}}{4}, \quad (23)$$

where $|f''|_{\max}$ is the maximum of the absolute value of $f''(x)$ over the domain $[a, b]$.

Proof For any segment in a uniform segmentation of $f(x)$, the difference between the exact value of f and its linear piecewise approximation is

$$L_i(x) = f(x) - \left[\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} (x - x_i) + f(x_i) \right]. \quad (24)$$

We require that

$$|L_i(x)| \leq 2\varepsilon. \quad (25)$$

By a process similar to that used in Theorem 1,

$$L_i(x) = \frac{1}{2} f''(x_{\text{mid}}^{(i)}) \frac{\Delta_i^2}{4} (\alpha_i^2(x) - 1) + O(\Delta_i^2). \quad (26)$$

Let $|f''|$ assume its maximum in $[x_i, x_{i+1}]$ at x_i^* . Since $x_{\text{mid}}^{(i)} = x_i^* + O(\Delta_i)$, we have

$$|f''(x_{\text{mid}}^{(i)})| = |f''(x_i^*)| + O(\Delta_i). \quad (27)$$

(25) and (26) imply that

$$\frac{1}{2} \left[\max_{x \in [x_i, x_{i+1}]} |f''(x)| \right] \frac{\Delta_i^2}{4} |\alpha_i^2(x) - 1| + O(\Delta_i^3) \leq 2\varepsilon. \quad (28)$$

(28) can be written as

$$\frac{1}{16\varepsilon} \left[\max_{x \in [x_i, x_{i+1}]} |f''(x)| \right] |\alpha_i^2(x) - 1| + O\left(\frac{\Delta_i}{\varepsilon}\right) \leq \frac{1}{\Delta_i^2}. \quad (29)$$

Now, as in Theorem 1, $\frac{\Delta_i}{\varepsilon} = O(\varepsilon^{-1/2})$, so that (29) can be written as

$$\Delta_i^2 \leq \frac{1}{\frac{1}{16\varepsilon} \left[\max_{x \in [x_i, x_{i+1}]} |f''(x)| \right] |\alpha_i^2(x) - 1| + O(\varepsilon^{-1/2})}$$

or

$$\Delta_i^2 \leq \frac{16\varepsilon}{\left[\max_{x \in [x_i, x_{i+1}]} |f''(x)| \right] |\alpha_i^2(x) - 1| + O(\varepsilon^{1/2})}. \quad (30)$$

Now, the right side of (30) is an upper bound for Δ_i^2 , where Δ_i , $i = 0, 1, \dots, s-1$ are s segments covering $[a, b]$, in each of which $|L_i(x)| \leq 2\varepsilon$, $x_i \leq x \leq x_{i+1}$. We want a uniform segmentation of $[a, b]$ with the minimum Δ_i . So, we choose $\alpha_i = 0$ in (30), replace $\left[\max_{x \in [x_i, x_{i+1}]} |f''(x)| \right]$ by the maximum of $|f''(x)|$ over the entire domain, and take the square root. We use the equality sign in (30) to conclude that

$$\Delta_i = \frac{4\sqrt{\varepsilon}}{\sqrt{\max_{x \in [a, b]} |f''(x)|} (1 + O(\varepsilon^{1/2}))}.$$

Thus,

$$s(\varepsilon) = \frac{b-a}{\Delta_i} = \frac{d(\varepsilon)}{\sqrt{\varepsilon}}, \quad (31)$$

where

$$d(\varepsilon) = \frac{\sqrt{\max_{x \in [a, b]} |f''(x)|} (b-a) (1 + O(\varepsilon^{1/2}))}{4}.$$

Since

$$\lim_{\varepsilon \rightarrow 0} d(\varepsilon) = d = \frac{\sqrt{\max_{x \in [a, b]} |f''(x)|} (b-a)}{4},$$

we see that (31) implies that, for a uniform optimal segmentation

$$s(\varepsilon) \sim \frac{d}{\sqrt{\varepsilon}} \quad (\varepsilon \rightarrow 0). \quad (32)$$

■

REFERENCES

1. J. Detrey and F. de Dinechin, "Table-based polynomials for fast hardware function evaluation," *Proc. of the 16th Inter. Conf. on Application-Specific Systems, Architecture and Processors (ASAP-'05)*, July 2005, pp. 328-333.
2. F. de Dinechin and A. Tisserand, "Multipartite table methods," *IEEE Trans. on Computers*, Vol. 54, No. 3, March 2005, pp. 319-330.
3. D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a line or its caricature," *The Canadian Cartographer*, Vol. 10, No. 2, pp. 112-122, 1973.
4. H. Hassler and N. Takagi, "Function evaluation by table look-up and addition," *Proc. of the 12th IEEE Symp. on Computer Arithmetic (ARITH'95)*, pp. 10-16, July 1995.
5. T. Knudstrup, "A model for minimizing numeric function generator complexity and delay," *M.S. Thesis*, Naval Postgraduate School, Monterey, CA, (<http://bosun.nps.edu/uhtbin/cgisirsi.exe/9FojKTYeoF/SIRSI/257000018/60/3/X/BLASTOFF>) Dec. 2007.
6. J.-M. Muller, *Elementary Functions - Algorithms and Implementation*, Birkhäuser, Boston, 1997.
7. S. Nagayama, T. Sasao, and J. T. Butler, "Numerical function generators using LUT cascades," *IEEE Trans. on Computers*, Vol. 56, No. 6 June 2007, pp. 826-838.
8. S. Nagayama, T. Sasao, and J. T. Butler, "Design method of numerical function generators based on polynomial approximation for FPGA implementation," *10th Euromicro Conference on Digital System Design, Architecture, Methods, and Tools (DSD 2007)*, August 27-31, 2007 Lubeck, Germany, pp. 280-287.
9. T. Sasao, J. T. Butler, and M. D. Riedel, "Application of LUT cascades to numerical function generators," *The 12th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI2004)*, Oct. 18-19, 2004, Kanazawa, Japan, pp.422-429.
10. T. Sasao, S. Nagayama, and J. T. Butler, "Numerical function generators using LUT cascades", *IEEE Transactions on Computers*, vol. 56, No. 6, June 2007, pp. 826-838.
11. P.T.P. Tang, "Table lookup algorithms for elementary functions and their error analysis," *Proc. 10th IEEE Symp. Computer Arithmetic*, P. Konerup and D. W. Matula, eds., pp. 232-236, June 1991.
12. A. Tisserand, "High-performance hardware operators for polynomial evaluation," *Inter. Jour. High Performance Systems Architecture*, Vol. No. 1, 2007.